Aspect Based Sentiment Analysis for Users Review Dataset Using Deep Learning and BERT

Karan Arora[1], **Sarthak Arora**[2]

[1]Department of Computer Science, Chitkara University, Punjab, India

[2]Department of IT, Maharaja Agrasen Institute of Technology, IPU, New Delhi, India

Abstract: ***Sentiment Analysis is a crucial part in Natural Language Processing (NLP), Aiming to relate pre-defined labels/categories to a given text sentence or sequence. It is very well recognised not only in academia but also in the industry , giving real-time outputs via internet reviews on websites like Amazon, which can utilise the customer's opinions on their products and services. The assumption of this task is that the entire text has an all-inclusive polarity. In this paper we aim to do sentiment analysis with BERT on the Review Dataset Collected by us. We did annotation of the data (11237 sentences) in the preprocessing phase which is one of the most crucial parts of the process then we use the outputs for the model as inputs that will be implemented on the same. We describe three classes related to the sentence idea namely "Usefulness", "Explanation" and "Competence" and two classes for polarity namely "Positive" and "negative". The output we get is the detailed sentiment analysis of the input review on the basis of the classes mentioned.***

Keywords: Bidirectional Transformers, Sentiment Analysis, Machine Learning, Deep Learning, BERT

1. Introduction

Aspect-based Sentiment analysis (ABSA), aims to identify a clear vision polarity in relation to a particular aspect, is a small challenging task of Sentiment Analysis. However, the Comments might contain various aspects, like: "The shirt looks sturdy but It is too costly" The polarity in 'appearance' is positive(+ve) , and the polarity in 'price' is negative(-ve). Proposed by (Jo and Oh, 2011; Pontiki et al., 2014, 2015 2016) Aspect-based sentiment analysis (ABSA) targets to find fine-grained polarity towards a particular aspect. It allows users to access collected sentiments for every aspect of a given product or service and obtain a better granular understanding of the product or service quality.

Both Sentiment Analysis(SA) and Aspect based sentiment Analysis (ABSA) are Sentence-level or Page-level tasks, but one comment might address more than one object, and sentence-level tasks can not work with sentences with multiple targets. For this Saeidi et al. (2016) introduce targeted aspect-based sentiment analysis (TABSA) task , which aims to find fine-grained opinion polarity towards a particular aspect linked with a given target. This work can be divided into two steps: (1) the first step is to identify the factors that correspond to the target; (2) The second step focuses on resolving the polarity to reach a given target. The Traditional Sentiment analysis focuses on distinguishing the general feeling expressed in the text without specifying the sentiment. This may not be enough if the text at the same time refers to different topics or things (Aspects), perhaps expressing the opposite emotions that point to different aspects.

Identifying sentiments related to various features in the text is a more complex task known as aspect-based sentiment analysis (ABSA). ABSA as a research topic received special accolades during SemEval-2014 (Pontiki et al., 2014) workshop, where it was first introduced as Task 4 re-appeared in SemEval-2015 (Pontet al., 2015) and SemEval-2016 (Pontiki et al., 2016) workshops.

2. BERT Induction

The Main Method is further discussed in section 4. This section provides an overview of the few strategies and models used throughout the remainder of the paper, as well as existing technical results. Section 2.1 will document a pre-trained model used in this paper, which has achieved state-of-the-art results in many NLP activities, as well as the structure of the model and its key features.

2.1 BERT

Pre-trained language models provide the context of the words, which they have previously studied occurrence and presentation of words from unselected training data. Bidirectional encoder representations from transformers (BERT) is a model designed to look at the context of name from left and right side simultaneously (Devlinet al., 2019). While the idea is simple, it improves results in many NLP activities such as emotional analysis and question and answer systems.

BERT can produce more contextual features in sequence compared to left and right training separately, like other such models as ELMo do (Peters et al., 2018). Pre-left and right training for BERT accessed using translated language masks, called a masked language model (MLM). The purpose of MLM is to hide random words in a sentence which are less probable. Model uses a Token to mask the word [MASK]. The model later tries to predict masked words from the left and to the right of the Masked word with the help of transformers. In addition to the left and right context domain uses MLM, BERT has a more important purpose that differs from previous works, namely the prediction of the following sentence.

2.2 Previous work

BERT is the state-of-the-art unsupervised model that is deeply bidirectional. There have been some previously trained language models before BERT also used unsupervised learning and were bidirectional. One of them ELMo (Peters et al., 2018), which too focuses on the content presentation of the content. Name ELMo embedding is generated using the Recurrent Neural Network (RNN) named Short-Term Memory (LSTM) (Sak et al., 2014) training from left to right and right to left independently and later combines both word presentations (Peters et al., 2018). BERT does not use LSTM to locate word context features, however instead it utilizes transformers (Vaswani et al., 2017), that are attention-based systems which do not use recurrence.

2.3 Input Representation

Text input for BERT model for the first time processed through a process called wordpiece tokenization (Wu et al., 2016). This produces a set of tokens, each representing a word. There are also two special tokens set of tokens: classification token[CLS], viz added at the beginning of the set; and the separating token [SEP], which indicates the end of a sentence. If BERT is used to compare the two sets of sentences, these sentences will be divided into a [SEP] token. This set of tokens is being processed later through three layers of embedding with the same size later summed together and transferred to the encoding layer: Token Embedding Layer, Partial Embedding Layer.

3. Data

The data was collected from a plethora of domains to maintain the real time complexity and challenges that the model must be evaluating on the test set. Most of the data was collected from top 100 sources collected carefully for diversity.

3.1 Data Collection Overview

The data consisted of people's reviews from top digital platforms in India like webstores , restaurants and general stores. Data Collection started in early - 2020 from Feb to April. To download the data , A scattered Crawler Framework was developed. The list of websites to gather were manually selected at first. We maintained a limit to the crawl hit so that the servers are not overloaded with requests thus data collection took several weeks. We collected reviews, ratings and additional reviews concerning a product or service as demonstrated on Table 1.

Table 1: Statistics for Collected Data

	Digital Products	Edibles	Internet Services
Men/Women	71/29 %	53/47 %	No Data
Rating System (Best To Worst)	1-5	1-5	1-10
Avg. Character Length	488	383	159
Review Texts	19,56,649	84,875	8,547

3.2 Rating Categories

The categories to be explained were identified on the basis of quantitative models. Here we used available categories that were labelled with the dataset i.e Positive and Negative. Rating categories were assigned after systematically merging to the set of categories decided after analysing. We will discuss the method later in section 3.3. The three decided classes were analyzed and categorised clearly before we discuss the annotation process in detail:

"Usefulness" class grades the amount of satisfaction the Product/ Service person has on the basis of the review written by him. Eg - If the positive adjective is linked to word like "very" then the review is considered as useful for the person , On the contrary when the negative adjective is linked to a word like "very", It is considered as negative. The "positive" and "negative" are clustered along with the classes and then passed on to the analysis process.

"Competence" - This class aims to justify the authority and originality of the person writing the review. Whether the person has actually bought and received the product or service, How long he received the product or service. How many times has he bought the product etc.

"Explanation" - How well the words in the sentence is used, How well the sentence is sequenced to make the case is the aim of this class. Eg - If the user is dissatisfied with the product , How many words has he written about the same and if he is extremely satisfied how well has he praised the product. The classes can be distinguished easily. In most cases, A multi word phrase needs to be annotated, Only nouns or singular words do not indicate a category. However there are some boundary cases in our data because of the complexity of some erroneous statements submitted by users because of which the annotation process is complicated.

3.3 Annotation Process

The process began with separating review sentences using spaCy library (ExplosionAI, 2019). Annotation at the sentence level instead of Document quality works well, especially if having features represented by complex phrases. This moreover, Pontiki et al., (2016b) also explained in sentence level. However, we have more than 2 millions of sentences for review.

Then we explain 10,000 sentences to show it whether they contain a statement that can be evaluated. Based on this case, after obtaining a higher agreement among the annotations, we built a Convolutional Neural Network (CNN) classifier in order to calculate a probability for every sentence to determine whether it contains a potential evaluation part or not. We have kept all the sentences with more than 50% random chances on file and used them as an adjective for certain sentences and their categories. We took this for granted compared to the use of seed names as performed by other scholars (Cieliebak et al., 2017).

Our vocabulary, when it comes to the description of aspect classes, is complex and often consists of longer phrases. Thus, we expect a narrowed selection of sentences when using seed words for our dataset. at least one of the three classes, 4,900 did not. You may describe several aspects in one sentence. Our annotations are stored in a database and are available sent to text files for further processing. Token making is stored here, too. The average sentence the length of the tokens can be found in Figure 1. Most sentences are shorter, while there is a certain number of long ones.

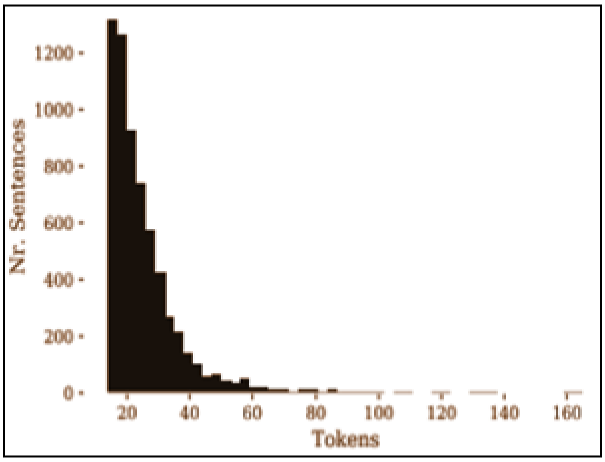

Figure 1: Number of token from sentences

The following sentence is a good example based on with sentences from the database: “speedy“ [Fast] and “sturdy” [Strong], are a good match: “The boot time was **speedy[Fast]** and the feel of the phone when i got it in my hands felt **sturdy [strong]** ” Feature names are printed with in bold. Here, for example it conveys the idea of common phrases. Most users write the way they talk. They measure the same object with long phrases or short words - usually not nouns - even several times in the same sentence. This differs in comparison to Pontiki et al. (2016b) also UWojatzki et al. (2017). However, compared to Pontiac Et al. (2016b), Our database is large. In addition, Pontiki et al. (2016a) include one of the possibilities it is mentioned several times with the same targeted view or feature organizations. Wojatzki et al. (2017), however, it seemed great at first to us, but to slow it down displays sentences with the characteristics of a defined feature that it is just a little bigger than ours (about 2,000 more sentences).

Annotation function was critical due to the nature of our data and calculated inter-annotator agreement on the basis of tagging, that is, all words found a tag with its own category and all missing words defined and marked “No class” (see Section 4). We randomly selected 337 (3%) of data defined by the main annotation before. After that, the other two, re-annotated them from scratch. We got enough contract points as can be seen in Table 3. We counted Cohen's Kappa (Cohen 1960) with two of the three adjectives used in Scikit-read (Pedregosa et al., 2011). Agreement between annotations there is a minimum of 0.722 and the magnitude between R and J is 0.857. According to Landis and Koch (1977), all prices between 0.61 and 0.80 can be considered as substantial, prices more than 0.81 as approx is perfect. We consider these results to be good for our dataset. We added Krippendorf's Alpha (Krippendorff, 2011) which makes use of NLTK (Bird et al., 2009) at the same time on all three annotations. Here, we get 0.771 points that may look good, where 1.0 would be much better. Alpha provides several benefits such as counting for many simultaneous annotations (not just two). Missing data too any category number can also be used (Krippendorff, 2011).

Table 2: Inter-annotator agreement between annotators R, B and J

Annotators	R & B	R & J	B & J
Cohen’s Kappa	0.722	0.857	0.73
Krippendorf’s	0.771	0.771	0.771

4. Methodology

In this section, we briefly describe our approach to make the feature phrase once classification of sentences on the basis of a defined database. At first, we explain the methods we followed by a search for a working system. We scanned the literature for building the ideal extraction system. For example, Liu (2012) suggests four ways to extract features: Extraction (1) using a common noun (phrases), (2) by making use of opinion and target relations (3) by supervised learning (4) based on the topic modeling (Liu, 2012). We tried and had to conclude that only supervised methods were promising. This is based on test results and a section of related books. For example, title modeling did not find subjects that were clearly categorized as people would explain themselves. Nouns often lead to extremely low detection rate of features and relationship withdrawals did not produce usable results. We used spaCy (ExplosionAI, 2019) for dependency parsing and results of Kitaev and Klein (2018) for constituency parsing to find candidate phrases. After several machine learning architectures for IOB tagging we found our approach to be superior.

The literature showed the superiority of IOB tagging (De Clercq et al., 2017). This does not seem right for our case,as we have long phrases with differing start words that may not be as predictable as in named entities. However, while IOB tagging does not fit, the idea is sufficient when leaving out the Beginning (B) tag in favor of only I and O. We tried both and the binary IO tagging proved to be the best solution.

When it comes to successive labeling activities, studies suggest using a Conditional Random Field (CRF) in combination with a bidirectional Recurrent Neural Network (RNN) (Toh and Su, 2016) which scrapes features, which we have done. We didn't use any extra features as mentioned by other scholars, e.g. named entity information or token lemmas, as we rely on user-generated content that has too many mistakes and nouns are not dominant for us. Still, the tests with Part-of-Speech tags and other common features did not improve our results. Architecture of our model can be seen in Figure 2.

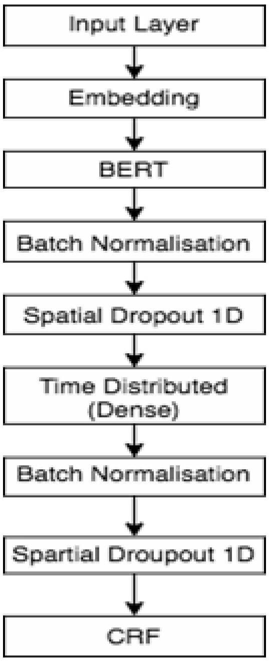

Figure 2: Model Architecture

Our model builds primarily on Bidirectional Encoder Representations from Transformers BERT (Devlin et al., 2018) to extract features in consecutive text data in both directions, using words before and after the current one. A time-distributed dense layer aligns all those features, before we hand them over to a CRF that considers the whole sentence in order to assign tags. The "BatchNormalization" layers are meant to keep the activation smaller, i.e., normalized. The dropout layers are used to prevent overfitting as our manually annotated dataset is relatively small. The input consists of sentences whose tokens were vectorized. At first, we used our tokens together with their tags in the form of "Positive" or "O" for a non-relevant word.

That is the reason, we directly trained the system for detecting aspect phrases together with their category. Secondly, it was crucial to have pretrained vectors.

We trained our vectors on all of our sentences with further measures for avoiding incorrectly split words and using only lowercase. Interestingly, the vectors that have size 300 worked very good for the model. This vector dimensionality helps to avoid overfitting as well increasing recall, especially in comparison to a small dimension size as 25.

The Embedding layer on Figure 1 contains all the vectors. We trained our own vectors using FastText (Bojanowski et al., 2017). As User-generated content contains many errors, we reduce it to minimize such errors. Our embedding is enriched with subword information (character nGrams), which are helpful when dealing with user-generated content to cover errors. We used the skip gram algorithm proposed by Bojanowki et al., (2017). It learns vector representations of words which can predict words appearing in the context. Use it time for parameter adjustment and testing of other model facilities using CNN, multiple RNN layers, more types of RNNs, models without CRF layers, etc. parameters showed positive results with values such as a out of 0.3, a small size of 30 units in LSTM layer, RMSprop as optimizer, a small epoch size due to a small dataset and a batch size of about 10.

5. Evaluation and Discussion

Table 2 presents our evaluation results such as precision, recall, F1-score per label as well as accuracy and an average per measure. While our accuracy of 0.95 is high, we regard our F1-score as more important. The F1-value of 0.80 is unweighted and can be regarded as good, especially in comparison to results in Pontiki et al., (2016b) or Wojatzki et al., (2017) who barely reach values of 0.50 in a domain with less complex wording and language while they separate extraction of phrases and classification of them which leads to forward propagation of errors. We also did this in order to not get overlapping aspect phrases for different categories.

Table 3: Evaluation results of our model (self-trained and BERT embeddings

Measures	P	R	F1	P(B)	R(B)	F1(B)
I-Explanation	0.81	0.71	0.76	0.73	0.67	0.7
I-Usefulness	0.75	0.74	0.75	0.75	0.69	0.72
I-Competence	0.61	0.68	0.67	0.67	0.65	0.67
WP-Time	0.85	0.8	0.82	0.87	0.77	0.82
O	0.97	0.98	0.97	0.97	0.98	0.97
Average	-	-	0.95	-	-	0.94
Accuracy	0.81	0.78	0.8	0.8	0.75	0.78

P = Precision, R = Recall, F1 = F1-score, B = BERT

However, our precision scores are usually better than our recall scores. We think this is coming from a very small amount of annotated training data. During training, overfitting was a problem and so on it was a goal to improve recall: We want our model to apply to new data and have good results.

Our BERT embedding achieves better memory and everything scores: Current memory values of 0.67 to 0.80 (also 0.98 of the label "O") are considered desirable, especially considering the F1- points of 0.76, 0.75, 0.67, 0.82 and 0.97 is more than it is satisfactory when considering the background and data. The accuracy indicates the highest value of 0.95 can be explained by the fact that usually, the label "O" appears more frequently and thus increases the accuracy points, Thus we relate to F1. Moreover, we take it as it is important that precision and recall are not too far apart. This is the reason why we prefer our model with embedded layout with highly trained word vectors BERT veters. As Table 4 reveals, BERT embedding (Devlin et al., 2018) empowers our model to achieve 0.67 points for "Competence", according to our embedding. While on average, precision scores are 0.80 compared to 0.81, recall is down 0.75 to 0.78. This is why we prefer our model which, as we think, also reflects our user generated data better.

To discuss our evaluation scores, it can be said that direct comparison with other models and subjects is not possible. This comes from the database we built and presented earlier in the third section. However, a comparison as indicated to the commonly presented values in studies dealing with shared tasks and their numerous results achieved in them indicates the superiority of our approach. While the IO Tags combined BERT-CRF model proved to be quite successful, self-trained word vectors finally enable test scores in Table 2.

Numerical scores can be misleading. Therefore, we regard manual tests as important. We have written many sentences that we take as edge-cases and cases that may be difficult to distinguish from overall. However, aspect extraction and the separation made by our model is more than satisfying. In addition, we have annotated higher database effects of inter-annotator agreement scores. We have used a few human resources, we gained the comparable as Cohen's Kappa scores to Wojatzki et al., (2017), even though they do not clearly depict scores for the aspect spans. Their inter-annotator agreement for aspects lies between 0.79 and 1.0. Pontiki et al., (2016b) use the F1-score for the annotator agreement. We consider this score to be tough to compare.

6. Conclusion

Earlier, we introduced the ABSA topic. Here we are shown current insufficient issues targeted. In addition, we have improved this understanding in the literature section by presentation current methods and general ideas from the area of ABSA. However, there is still much work to be done to extend research from standard reviews of integration products and resources for more language complex review areas and languages, prior ABSA may work for other domains. After that, we introduce our own data. Here, we have collected a large number of reviews texts to use train word embedding and subtract sentences that had been explained. We describe three classes related to the sentence idea namely “Usefulness” , “Explanation” and “Competence”. Our currently defined database has 11,237 sentences.

We plan to continue extending the process to other sections and ideas purposes such as the main search page and the product description. We also provided examples and comparisons with other data sets, details named by segmentation and calculated the inter-annotator agreement so that you get good points. After that we explained our method of extraction again to distinguish feature phrases. This includes the model construction process and parameter process tuning and details about word embedding training. However, we say what did not lead to success as well as difficulties for aspect extraction performed by a machine learning system

Finally, we evaluate and discuss our performance scores. Here, we compare our model with others. On the contrary for some scholars, the way we work makes two steps in one: paragraph clause and subdivision. However, we go beyond other such approaches as Pontiki et al., (2016b) even though we use another domain and database. However, we are looking at this domain as a complex using the morally rich sentences. In the future, we are not just planning to build more data sets, but you want to enter a view part of the output, too.

References

[1] Erik Cambria, Soujanya Poria, Rajiv Bajpai, and Bjorn ̈ Schuller. 2016. Senticnet 4: A semantic resource for sentiment analysis based on conceptual primitives. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2666–2677.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[3] Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2016. Tracking the world state with recurrent entity networks. arXiv preprint arXiv:1612.03969.

[4] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2015. Effective lstms for target dependent sentiment classification. arXiv preprint arXiv:1512.01100.

[5] Bäumer, F. S.; Grote, N.; Kersting, J.; and Geierhos, M. 2017. Privacy Matters: Detecting Nocuous Patient Data Exposure In Online Physician Reviews. In Proceedings of the 23rd International Conference on Information and Software Technologies,, 756, 77–89. Druskininkai, Lithuania: Springer.

[6] Bäumer, F. S.; Kersting, J.; Kuršelis, V.; and Geierhos, M. 2018. Rate Your Physician: Findings From A Lithuanian Physician Rating Website. In Proceedings of the 24th International Conference on Information and Software Technologies, Communications in Computer and Information Science, 920, 43–58. Vilnius, Lithuania: Springer.

[7] Danda, P.; Mishra, P.; Kanneganti, S.; and Lanka, S. 2017. IIIT-H at IJCNLP-2017 Task 4: Customer Feedback Analysis Using Machine Learning and Neural Network Approaches. In Proceedings of the 8th International Joint Conference on Natural Language Processing, Shared Tasks, 155–160. Taipei, Taiwan: AFNLP.

[8] De Clercq, O.; Lefever, E.; Jacobs, G.; Carpels, T.; and Hoste, V. 2017. Towards an Integrated Pipeline for Aspect-based Sentiment Analysis In Various Domains. In Proceedings of the 8th ACL Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis. 136–142, Copenhagen, Denmark: ACL.

[9] Emmert, M.; Sander, U.; and Pisch, F. 2013. Eight Questions About Physician-Rating Websites: A Systematic Review. Journal of Medical Internet Research 15(2):e24.

[10] Kersting, J.; Bäumer, F.; and Geierhos, M. 2019. In Reviews We Trust: But Should We? Experiences With Physician Review Websites. In Proceedings of the 4th International Conference on Internet of Things, Big Data and Security, 147–155. Heraklion, Greece: SCITEPRESS.

[11] Krippendorff, K. (2011). Computing Krippendorff’s AlphaReliability. University of Pennsylvania. Retrieved from https://repository.upenn.edu/asc_papers/43

[12] Zeithaml, V. 1981. How Consumer Evaluation Processes Differ Between Goods and Services. Marketing of Services 9(1):186–190.

[13] Nguyen, T. H., and Shirai, K. 2015. PhraseRNN: Phrase Recursive Neural Network for Aspect-based Sentiment Analysis. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2509–2514. Lisbon, Portugal: ACL.

[14] Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; AL-Smadi, M.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq , O.; Hoste, V.; Apidianaki, M.; Tannier, X.; Loukachevitch, N.; Kotelnikov, E.; Bel, N.; Jime ́nez-Zafra, S. M.; and Eryig it, G. 2016b. Semeval-2016 Task 5: Aspect Based Sentiment Analysis. In Proceedings of the 10th International Workshop on Semantic Evaluation, 19–30. Denver, CO, USA: ACL.

[15] Chinsha, T. C., and Shibily, J. 2015. A Syntactic Approach for Aspect Based Opinion Mining. In Proceedings of the 9th IEEE International Conference on Semantic Computing. 24–31. Anaheim, CA, USA: IEEE.

[16] Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. In Proceedings of the 32rd AAAI Conference on Artificial Intelligence (AAAI 2018). New Orleans, USA.

[17] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017). Copenhagen, Denmark, pages 452–461.